

Etape 10 : Suspendre un Thread JAVA

Multithreading en JAVA

Suspension : interrompre temporairement le thread

- Il est parfois utile d'interrompre momentanément *un thread*
- On va d'abord se focaliser sur la *suspension temporelle* et voir TOUTES les conséquences
- Java appelle cela la mise en sommeil accessible par la méthode *sleep()* :

=> **Thread.sleep(durée)**

- ▶ La durée peut s'exprimer en millisecondes ou nanosecondes :
 - ▶ `sleep(nombre de millisecondes, nombre de nanosecondes)`

Interruption du sommeil d'un *thread*

- ▶ Remarque 1 : elle est susceptible de déclencher "*InterruptedException*"
- *Un thread n'aime pas être réveillé dans son sommeil !*
- La suspension d'un thread place le thread dans un état particulier (nommé *TIME-WAITING*) où il devient vulnérable à une demande d'interruption externe
 - cela conduit logiquement au déclenchement de l'exception :
InterruptedException
 - S'il elle n'est pas géré par le thread cela peut conduire à la fin du Thread

Suspension d'un *thread*

- Remarque 2 : sleep est une *méthode statique* !
- *mais alors ... qui s'endort ?*
 - c'est le thread où la méthode est invoquée qui se met en sommeil !
 - => conforme au principe d'autogestion des thread Java : le thread gère son cycle de vie (et non pas un thread externe)

Suspension d'un *thread*

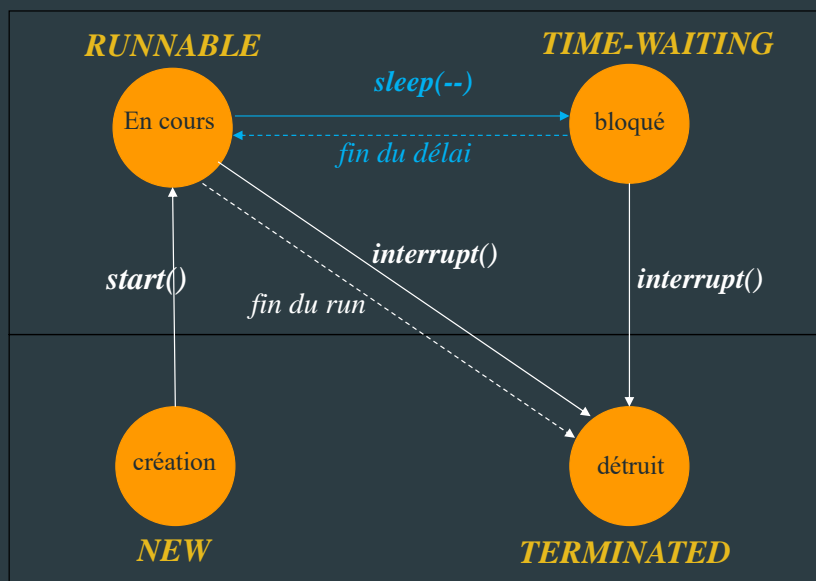
```
class MonThread extends Thread {
    public void run() {
        try {
            System.out.println("Début du thread, attente de 7 secondes...");
            Thread.sleep(7000);
            System.out.println("7 secondes se sont écoulées.");
        } catch (InterruptedException ie) {
            System.out.println("Réveillé avant la fin du délai !");
        }
    }

    public static void main(String[] args) {
        try {
            MonThread t = new MonThread();
            t.start();
            Thread.sleep(3000);
            System.out.println("On interrompt le thread t après seulement 3s.");
            t.interrupt();
        } catch (InterruptedException ie) {ie.printStackTrace();}<
    }
}
```

en cas
d'interruption
pendant le
sommeil

Le *thread* dans tous ses états

- la méthode `getState()` donne l'état du thread sur lequel elle est appliquée :



la JVM accorde du temps
processeur à un *thread* :

`isAlive()==True`

`isAlive()==False`